

The background of the entire slide is a vibrant green. On the right side, there is a wireframe profile of a human head, facing left. The head is composed of a grid of thin, light-green lines. Behind the head, there are faint, glowing green circuit patterns and some electronic components like capacitors and resistors, some labeled with codes like 'C128', 'CM30', 'CM28', 'CM42', 'CM43', 'CM44', 'CM45', 'CM46', 'CM47', 'CM48', 'CM49', 'CM50', 'CM51', 'CM52', 'CM53', 'CM54', 'CM55', 'CM56', 'CM57', 'CM58', 'CM59', 'CM60', 'CM61', 'CM62', 'CM63', 'CM64', 'CM65', 'CM66', 'CM67', 'CM68', 'CM69', 'CM70', 'CM71', 'CM72', 'CM73', 'CM74', 'CM75', 'CM76', 'CM77', 'CM78', 'CM79', 'CM80', 'CM81', 'CM82', 'CM83', 'CM84', 'CM85', 'CM86', 'CM87', 'CM88', 'CM89', 'CM90', 'CM91', 'CM92', 'CM93', 'CM94', 'CM95', 'CM96', 'CM97', 'CM98', 'CM99', 'CM100'.

## **Lección 02:**

### **Estructuras de control secuenciales**

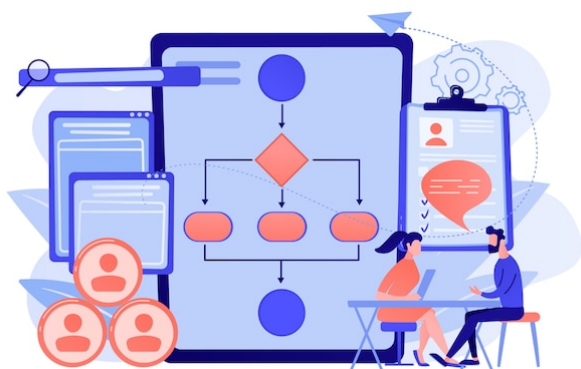
Fredy Camacho García  
Jose Alonso Oviedo  
Andrés Mauricio Arciniegas

# 2

## Algoritmos

Un algoritmo es una secuencia ordenada de tareas o pasos, ordenados de forma lógica, que permiten establecer la solución a un problema.

En la presente lección se emplearán herramientas computacionales tales como lenguajes de programación y entornos de desarrollo para construir un algoritmo.



En un algoritmo se identifican tres elementos:

- **Entrada:** Conjunto de datos necesarios para que el algoritmo se pueda ejecutar a cabo con éxito.
- **Proceso:** Conjunto de pasos por llevar a cabo. Los datos de entrada son tomados y transformados en resultados.
- **Salida:** Resultados esperados después de ejecutar el proceso. Puede ser otro conjunto de datos o información.

Todo algoritmo debe ser:

- **Finito:** La cantidad de acciones debe terminar en algún momento.
- **Preciso:** Si se proporciona siempre los mismos datos de entrada al algoritmo, se deben generar los mismos resultados.

**Ejemplo 1:** Identificar los elementos de un algoritmo para preparar una receta de cocina

### Solución

**Entrada:** Ingredientes del plato a preparar

**Proceso:** Preparación donde se especifica cada una de las tareas a realizar en un orden lógico

**Salida:** Plato preparado

## Herramientas computacionales

**Lenguaje de programación:** Se utilizará Python como lenguaje de programación para aprender a construir algoritmos. Recuerde que un lenguaje de programación es:

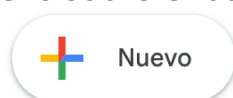
- Un programa que permiten crear otros programas
- Para tal fin, se lleva a cabo la escritura de código utilizando algún editor o entorno de desarrollo.
- El código debe cumplir con una sintaxis propia, esto es, un conjunto de reglas que definen la forma en que se debe escribir el código.

**Google colabory:** Llevará a cabo la función de entorno de desarrollo. Allí puede escribir en un documento de texto el código que compone a su algoritmo, utilizando la sintaxis de Python y, posteriormente, ejecutarlo para verificar su funcionamiento.

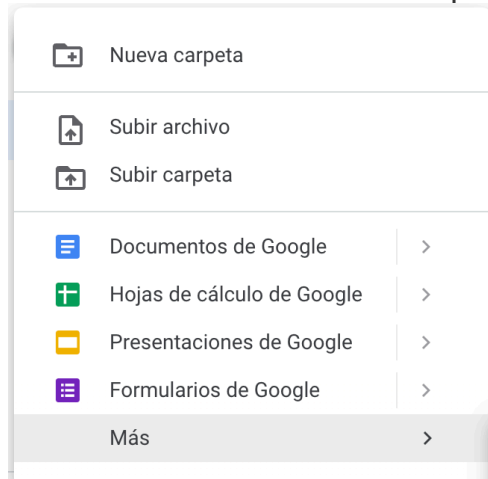
Para ingresar a **Google Colaboratory** existen dos formas:

La primera es a través de una cuenta basada en **gMail**, mediante la aplicación **Google Drive**, creando un documento nuevo de tipo Google Colaboratory, de la siguiente forma:

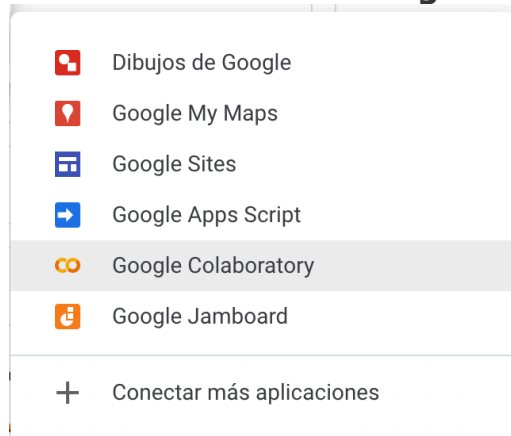
Clic sobre el botón **Nuevo**



Seleccionar en el menú desplegable la opción **Más**



Hacer clic en el ítem **Google Colaboratory**



La segunda forma es ingresando a través de un navegador web de forma directa, escribiendo en la barra de direcciones el enlace <https://colab.research.google.com/>

## El primer programa



El primer programa, por lo general, consiste en mostrar un mensaje en pantalla.

Para mostrar un mensaje se utiliza la instrucción **print()**

Sintaxis:

```
print("el mensaje por mostrar")
```

La instrucción **print()** siempre se escribe en minúscula (y generalmente en Python todas las instrucciones se escriben en minúscula). Esto se conoce como caso sensitivo, dado que, se diferencia las mayúsculas de las minúsculas.

Para escribir el código, deberá pulsar el botón **Bloc de notas nuevo**. Una vez se abra el editor de código, se pulsará el botón **+ Código** y se escribirán las instrucciones que componen al programa.

### Ejemplo 1.




Mostrar un mensaje en pantalla

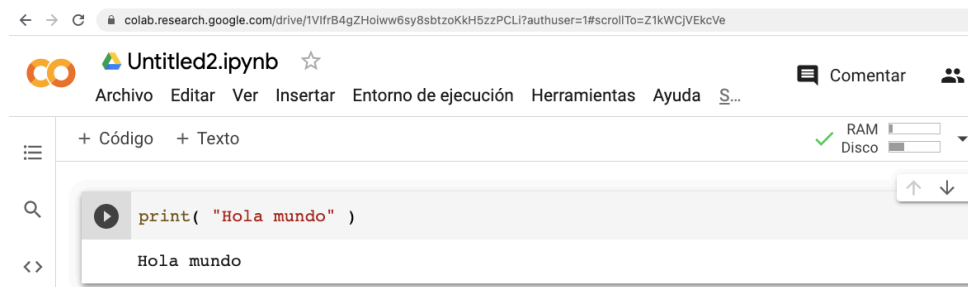
### Solución

Se deberá ingresar el siguiente código en el editor:

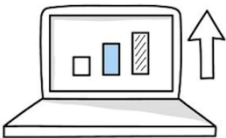
```
print( "Hola mundo" )
```

Para ejecutar se pulsa el botón  ubicado sobre la izquierda y el resultado de la ejecución se observará debajo del código escrito.

Así se debe observar el entorno de desarrollo:



## Ejemplo 2.




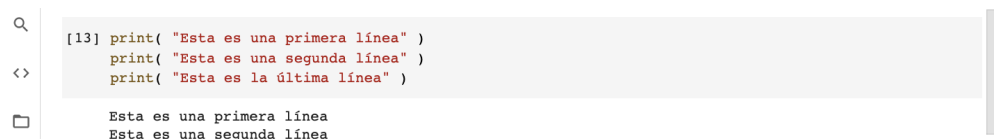
Mostrar dos mensajes en pantalla, uno a continuación del otro.

## Solución

Se pulsa de nuevo el botón **+ Código** y se ingresa las siguientes instrucciones:

```
print( "Esta es una primera línea" )  
print( "Esta es una segunda línea" )
```

Se ejecuta la celda de código pulsando el botón . El resultado es el siguiente:





## Los datos



Un dato es un valor que representa algo del mundo real.

Por ejemplo, 1.75 es un dato, el cual es de tipo numérico. El significado se lo da el programador. Puede ser la estatura en metros de una persona, puede ser la distancia en Km desde el lugar donde se encuentra alguien hasta una ubicación específica que se requiere, el valor en dólares de un dulce, o también, el volumen medido en litros de una bebida gaseosa, entre otros.

Otro ejemplo: 30 es un dato, el cual es de tipo numérico.

Por ejemplo:

1.75	Estatura en metros de una persona Distancia en kilómetros entre dos puntos Valor en dólares de un dulce
------	---

La interpretación del valor (contexto en donde este se presenta) se la da la persona que posee dicho dato.

30	Edad en años de una persona Número de horas de un curso Cantidad de estudiantes matriculados en un curso
----	--

"Jorge"	Nombre de una persona Nombre de una caricatura famosa de tv Seudónimo de un poeta
---------	---

Existen valores cuya naturaleza es numérica: 1.75, 30

Existen valores cuya naturaleza es alfanumérica: "Jorge"

Lo anterior significa que existen diferentes **tipos de datos**.

## Tipos de datos



### Numéricos:

Los valores son números. Pueden ser:

- **Entero:** Valores numéricos sin decimales, por ejemplo, 30
- **Real:** Valores numéricos que pueden manejar dígitos decimales, por ejemplo, 1.75

### Alfanuméricos:

Los valores incluyen letras, dígitos y símbolos (caracteres). Pueden ser:

- **Carácter:** Un carácter es cualquier símbolo que la computadora puede reconocer. Está representado por un único símbolo. Por ejemplo, el sexo de una persona en su documento de identidad que puede ser 'F' o 'M'. Por lo general, se encierran entre apóstrofes (comilla simple).
- **Cadena:** Es un conjunto de caracteres agrupados como un único valor. Por ejemplo, el nombre de una persona: "Juan Pérez". Por lo general, se encierran entre comillas dobles.

### Lógicos:

Valores que representan un valor de verdad: Verdadero (True) o Falso (False). Por ejemplo, el estado de un estudiante en una institución académica: Activo (True) o Inactivo (False).

En programación, el separador decimal siempre es **el punto**. De otra parte, los datos deben tener un nombre, es decir, un identificador que permita utilizarlos en cualquier parte del programa y hacer referencia al mismo dato.

De acuerdo con su naturaleza, los datos pueden ser:

- **Variables.** Representan cualquier valor, y el dato puede cambiar las veces que quiera de valor
- **Constantes.** El valor nunca cambia (En python no se manejan constantes).

### Ejemplo 3.



Crear un script que defina una variable llamada x y su valor es 5.

### Solución

```

x = 5
print( x )
5

```

En el script anterior, se define una variable con identificador **x** a la cual se le asigna el valor **5**. Posteriormente, se muestra en pantalla (imprime) el valor de la variable x.

Observe que la variable x no está encerrada entre comillas. Si se hace, se imprime como un mensaje el texto "x", es decir, se muestra el caracter x en pantalla mas no el valor de la variable x. En la siguiente figura se puede apreciar este error muy común

```

x = 5
print( "x" )
x

```

Ahora bien, si se requiere acompañar el valor de la variable de un mensaje, la instrucción **print()** permite que se despliegue varios elementos a la vez, separándolos por coma:





### Ejemplo 4.

Asignar el valor 10 a la variable `x` y mostrarlo en pantalla acompañado de un mensaje.

### Solución

```

x = 10
print( "El valor de x es:", x )

El valor de x es: 10

```

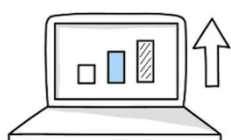
Algo interesante en programación es que las variables no se tienen que llamar `x` o `y` como ocurre en el álgebra. De hecho, no se recomienda hacer esto. Preferiblemente, establezca identificadores significativos, es decir, que al momento de leerlos se comprenda de forma inmediata lo que representa.



Los identificadores (nombres de datos) deben cumplir con las siguientes normas y reglas:

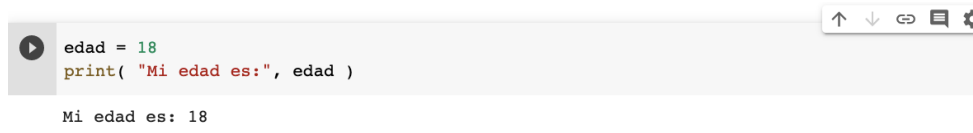
- Debe ser significativo (cuando se lea se interprete inmediatamente que representa)
- Debe comenzar con letra minúscula, preferiblemente
- Puede contener números, o guion bajo ( `_` )
- No puede contener espacios
- Si se requiere varias palabras, generalmente se utiliza la notación **Camel Case**, por ejemplo, el identificador *nombreCompleto*, la segunda, tercera y demás palabras simples comienzan en mayúsculas. Otros ejemplos, *salarioBase*, *porcentajeDeDescuento*, entre otros.

### Ejemplo 5.



Definir una variable `edad`, y asígnele su edad. Muestre en pantalla el mensaje: *Mi edad es XX* haciendo uso de la variable definida.

## Solución.



```
edad = 18
print( "Mi edad es:", edad )
```

Mi edad es: 18

Observe la instrucción **print()**, en la cual se especifica que se debe desplegar en pantalla dos valores: una cadena de caracteres con el valor "Mi edad es: ", y unido a esta, el valor de la variable *edad*, a la cual se le asignó en la instrucción anterior el valor 18.

## Ejemplo 6.



Definir 3 datos que representen su nombre, su edad y su estatura, desplegarlas en pantalla, donde cada uno esté acompañado de una cadena de caracteres que indique lo que representa.

Recuerde que los datos alfanuméricos tienen valores que se encierran entre comillas. Por ejemplo, *nombre* = "Juan"

## Solución



```
nombre = "Andrés"
edad = 30
estatura = 1.75
print( "Mi nombre es:", nombre )
print( "Mi edad es:", edad )
print( "Mi estatura es:", estatura )
```

Mi nombre es: Andrés  
Mi edad es: 30  
Mi estatura es: 1.75

Note que los datos numéricos tienen asignado un valor el cual no contiene delimitadores a diferencia de los datos alfanuméricos, los cuales se encierran entre comillas dobles.

## Operadores



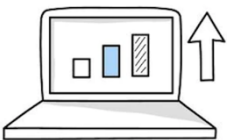
Es un símbolo que expresa una operación por realizar.

Existen diferentes tipos de operadores:

- **Aritméticos**
  - + Suma
  - - Resta
  - \* Producto
  - / División Real
  - \*\* Potenciación
  - // División Entera (Cociente)
  - % Residuo de la división entera
  - ( ) Paréntesis para agrupamientos
- Relacionales
- Lógicos
- Cadena

En esta lección se tratará únicamente los operadores aritméticos. Los demás tipos de operadores serán tratados en lecciones posteriores (por eso, se resaltan en color gris).

### Ejemplo 7.



Escribir un script que permita sumar dos números y desplegar el resultado en pantalla.

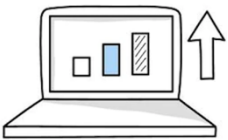
### Solución.

```
# Operadores aritméticos
numero1 = 2
numero2 = 5
resultado = numero1 + numero2
print(resultado)
```

## Resultado de la ejecución.

La suma es: 7

## Ejemplo 8.



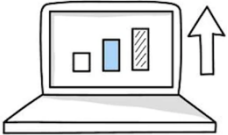
Escribir un script que permita realizar las cuatro operaciones básicas entre dos números y desplegar cada resultado en pantalla.

## Solución

```
# Operadores aritméticos (II)
numero1 = 10
numero2 = 4
suma = numero1 + numero2
resta = numero1 - numero2
producto = numero1 * numero2
cociente = numero1 / numero2
print( "La suma es:", suma )
print( "La resta es:", resta )
print( "La multiplicacion es:", producto )
print( "La división es:", cociente )
```

## Resultado de la ejecución.

La suma es: 14  
La resta es: 6  
La multiplicacion es: 40  
La división es: 2.5



### Ejemplo 9.

Escribir un script que permita realizar operaciones aritméticas de potenciación, división entera y residuo entre dos números y desplegar cada resultado en pantalla.

### Solución

```
# Operadores aritméticos (III)
numero1 = 7
numero2 = 3
potencia = numero1 ** numero2
division = numero1 // numero2
residuo = numero1 % numero2
print( "La potencia es:", potencia )
print( "La división entera es: ", division )
print( "El residuo de la división es:", residuo )
```

### Resultado de la ejecución.

```
La potencia es: 343
La división entera es:  2
El residuo de la división es: 1
```

## Lectura de datos

Consiste en solicitar al usuario (quien utiliza el programa, es decir, quien lo ejecuta) los valores de los datos de entrada necesarios para que el script se ejecute.

Para hacer lectura de datos se utiliza la instrucción **input()** con la siguiente sintaxis:

```
dato = input()
```

En este caso, la variable que está a la izquierda del igual recibe (se le asigna) un valor que es ingresado por teclado.

La instrucción **input()** ocasiona que el programa detenga su ejecución con el fin de que se muestre un espacio en blanco para que se ingrese un texto, y ese valor ingresado es el que toma la variable (dato) que está a la izquierda del igual.

### Ejemplo 10.

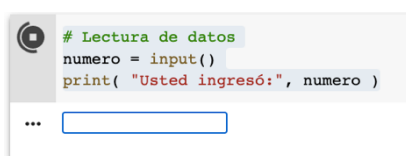


Leer un valor por teclado y mostrarlo en pantalla.

### Solución

```
# Lectura de datos
numero = input()
print( "Usted ingresó:", numero )
```

### Resultado de la ejecución.



```
10
Usted ingresó: 10
```

Si se quiere que para hacer la lectura de datos se despliegue un mensaje, este se puede colocar dentro de los paréntesis de la instrucción **input()** tal como se hace con **print()**. Por ejemplo:

```
# Lectura de datos
numero = input( "Por favor ingrese un número: " )
print( "Usted ingresó:", numero )
```



El resultado de la ejecución es el siguiente:

```
Por favor ingrese un número: hola
Usted ingresó: hola
```

### Ejemplo 11.



Modificar el ejemplo 7, en el cual se suman dos números para que estos valores sean ingresados por parte del usuario a través del teclado.

### Solución

```
numero1 = input( "Ingrese el primer número: ")
numero2 = input( "Ingrese el segundo número: ")
resultado = numero1 + numero2
print( "La suma es:", resultado )
```

### Resultado de la ejecución

```
Ingrese el primer número: 2
Ingrese el segundo número: 8
La suma es: 28
```

Lo anterior presenta un problema. En vez de hacer una suma aritmética, lo que ocurrió es que se unieron (concatenaron) los valores leídos. Esto ocurre porque un valor leído por teclado es un texto (mas no un número), y por tanto, el operador "+" actúa como concatenador de textos (cadenas).

Para solucionar lo anterior, se va a expresar los valores leídos como números enteros. Para ello se utiliza la instrucción **int()** de la siguiente forma:

```
dato = int( input() )
```

Por ejemplo,

```
numero = int( input() )
```

También lo puede acompañar de un mensaje:

```
numero = int( input( "Ingrese un número" ) )
```

Si los números por leer contienen decimales, se deben tratar como números de tipo real, por tanto, la instrucción por emplear es **float( )**.

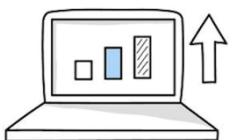
Así entonces, la solución al ejemplo es la siguiente:

```
numero1 = int( input( "Ingrese el primer número: " ) )  
numero2 = int( input( "Ingrese el segundo número: " ) )  
resultado = numero1 + numero2  
print( "La suma es:", resultado )
```

## Resultado de la ejecución

```
Ingrese el primer número: 4  
Ingrese el segundo número: 9  
La suma es: 13
```

## Ejemplo 12.



Modificar el ejemplo anterior, en el cual se suman dos números para que los valores ingresados por parte del usuario a través del teclado incluyan decimales.

## Solución

```
numero1 = float( input( "Ingrese el primer número: " ) )  
numero2 = float( input( "Ingrese el segundo número: " ) )  
resultado = numero1 + numero2  
print( "La suma es:", resultado )
```

## Resultado de la ejecución

Ingrese el primer número: 4.5  
 Ingrese el segundo número: 3.8  
 La suma es: 8.3

## Ejemplo 13.



Escribir un script que lea un número, calcule y visualice su doble.

### Solución.

Se identificará los datos necesarios para ser solicitados al usuario (datos de entrada) y los que representan los resultados por generar (datos de salida):

- *Datos de Entrada:* numero
- *Datos de Salida:* doble

```

# Ejercicio 1. Escribir un algoritmo que lea un número, calcule y visualice su doble

# Datos de Entrada: numero
# Datos de Salida: doble

# Entrada
numero = int( input( "Ingrese el número: " ) )

# Proceso
doble = numero * 2

# Salida
print( "El doble es:", doble )

Ingrese el número: 4
El doble es: 8
  
```

## Ejemplo 14.



Escribir un script que lea un número, calcule y visualice su cuadrado y su cubo.

## Solución

- *Datos de Entrada:* numero
- *Datos de Salida:* cuadrado, cubo

```
# Ejercicio 2. Escribir un algoritmo que lea un número, calcule y visualice su cuadrado y su cubo

# Datos de Entrada: numero
# Datos de Salida: cuadrado, cubo

# Entrada
numero = int( input( "Ingrese el número: " ) )

# Proceso
cuadrado = numero * numero
cubo = numero * numero * numero

# Salida
print( "El cuadrado es:", cuadrado )
print( "El cubo es: ", cubo )
```

➤ Ingrese el número: 5  
El cuadrado es: 25  
El cubo es: 125

Una segunda solución posible puede ser la siguiente:

```
# Ejercicio 2. Escribir un algoritmo que lea un número, calcule y visualice su cuadrado y su cubo

# Datos de Entrada: numero
# Datos de Salida: cuadrado, cubo

# Entrada
numero = int( input( "Ingrese el número: " ) )

# Proceso
cuadrado = numero ** 2
cubo = numero ** 3

# Salida
print( "El cuadrado es:", cuadrado )
print( "El cubo es: ", cubo )
```

➤ Ingrese el número: 4  
El cuadrado es: 16  
El cubo es: 64

Una alternativa a las propuestas anteriores también puede ser como la siguiente:

```
# Ejercicio 2. Escribir un algoritmo que lea un número, calcule y visualice su cuadrado y su cubo

# Datos de Entrada: numero
# Datos de Salida:  cuadrado, cubo

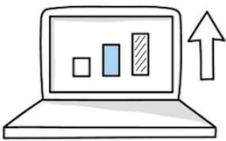
# Entrada
numero = int( input( "Ingrese el número: " ) )

# Proceso
cuadrado = numero ** 2
cubo = cuadrado * numero

# Salida
print( "El cuadrado es:", cuadrado )
print( "El cubo es: ", cubo )
```

Ingrese el número: 4  
 El cuadrado es: 16  
 El cubo es: 64

## Ejemplo 15.



Leer dos calificaciones de un estudiante y calcular su promedio. Las calificaciones estarán entre 0.0 y 5.0 (pueden contener decimales).

## Solución

- *Datos de Entrada:* nota1, nota2
- *Datos de Salida:* promedio

```
# 3. Leer dos calificaciones de un estudiante y calcular su promedio.
# Las calificaciones estarán entre 0.0 y 5.0 (pueden contener decimales).

# Datos de Entrada: calificacion1, calificacion2
# Datos de Salida: promedio

# Entrada
calificacion1 = float( input( "Ingrese la primera calificación: " ) )
calificacion2 = float( input( "Ingrese la segunda calificación: " ) )

# Proceso
promedio = ( calificacion1 + calificacion2 ) / 2

# Salida
print( "El promedio de las calificaciones es: ", promedio )
```

Ingrese la primera calificación: 3  
 Ingrese la segunda calificación: 4  
 El promedio de las calificaciones es: 3.5



Cuando una operación (técnicamente se conoce como **expresión**) incluye más de un operador, se presenta una precedencia, es decir, una priorización en la ejecución de estos.

La precedencia (jerarquía) de operadores es la siguiente:

Agrupamiento:	( )
Potenciación:	**
Multiplicativos:	*, /
Aditivos:	+, -

Los operadores ubicados por encima tienen mayor precedencia que los que se encuentren debajo.

Si en una expresión (operación) existen varios operadores de la misma precedencia, se ejecutan de izquierda a derecha.

## Expresiones



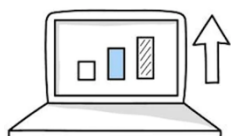
Una expresión es un conjunto de datos y operadores los cuales devuelven un resultado. Por tanto, las expresiones se clasifican según el resultado obtenido.

Las expresiones pueden ser:

- **Numéricas:** Cuando el resultado es un número, por ejemplo,  $3 + 5$
- **Alfanuméricas:** Cuando el resultado es una cadena de caracteres, por ejemplo, "buenos " + "días"
- **Lógicas:** Cuando el resultado es un valor lógico o booleano, por ejemplo,  $3 < 5$



### Ejemplo 16.



Escribir un script que calcule y visualice el resultado de la siguiente expresión

$$\frac{(a + b)^2}{3}$$

### Solución

- *Datos de Entrada:* a, b
- *Datos de Salida:* expresion

```
# 4. Escribir un algoritmo que calcule el resultado de la siguiente expresión:
# (a + b) ^ 2 / 3

# Datos de Entrada: a, b
# Datos de Salida: expresion

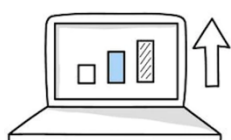
# Entrada
a = float( input( "Ingrese el valor de A: " ) )
b = float( input( "Ingrese el valor de B: " ) )

# Proceso
expresion = ( (a + b) ** 2 ) / 3

# Salida
print( "El resultado de la expresión (a + b)^2 / 3 es: ", expresion )
```

Ingrese el valor de A: 4  
 Ingrese el valor de B: 5  
 El resultado de la expresión (a + b)^2 / 3 es: 27.0

### Ejemplo 17.



Construya un script, tal que, dados como datos la base y la altura de un rectángulo, calcule y visualice la superficie y el perímetro de este.

### Solución

- *Datos de Entrada:* base, altura
- *Datos de Salida:* superficie, perimetro

Recuerde que,

$$\text{Área} = \text{base} * \text{altura}$$

$$\text{Perímetro} = \text{base} + \text{base} + \text{altura} + \text{altura}$$

o su equivalente,  $\text{Perímetro} = (\text{base} * 2) + (\text{altura} * 2)$

```
# 5. Construya un algoritmo, tal que, dados como datos la base y la altura de un
# rectángulo, calcule y visualice la superficie y el perímetro del mismo.

# Datos de Entrada: base, altura
# Datos de Salida: superficie, perímetro

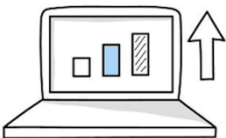
# Entrada
base = float( input( "Ingrese la base del rectángulo: " ) )
altura = float( input( "Ingrese la altura del rectángulo: " ) )

# Proceso
superficie = base * altura
perímetro = (base + altura) * 2

# Salida
print( "La superficie del rectángulo es: ", superficie )
print( "El perímetro del rectángulo es: ", perímetro )
```

Ingrese la base del rectángulo: 10  
 Ingrese la altura del rectángulo: 4.5  
 La superficie del rectángulo es: 45.0  
 El perímetro del rectángulo es: 29.0

## Ejemplo 18.



Construya un script, tal que, dados como datos el radio y la altura de un cilindro, calcule y visualice la superficie y el volumen de este. Considere un cilindro sin tapas.

## Solución

- *Datos de Entrada:* base, altura
- *Datos de Salida:* superficie, perímetro

Recuerde que,

$$\text{Superficie} = 2 * \pi * \text{radio} * \text{altura}$$

$$\text{Volumen} = \pi * \text{radio}^2 * \text{altura}$$

```
# 6. Construya un algoritmo, tal que, dados como datos el radio y la altura de
# un cilindro, calcule y visualice la superficie y el volumen del mismo.
# Considere un cilindro sin tapas.

# Datos de Entrada: radio, altura
# Datos de Salida: superficie, volumen

# Entrada
radio = float( input( "Ingrese el radio del cilindro: " ) )
altura = float( input( "Ingrese la altura del cilindro: " ) )

# Proceso
pi = 3.1416
superficie = 2 * pi * radio * altura
volumen = pi * radio ** 2 * altura

# Salida
print( "La superficie del cilindro es: ", superficie )
print( "El volumen del cilindro es: ", volumen )
```

Ingrese el radio del cilindro: 10  
 Ingrese la altura del cilindro: 2  
 La superficie del cilindro es: 125.664  
 El volumen del cilindro es: 628.3199999999999

## Ejercicios Propuestos



Construya un script tal que, dado el costo de un artículo vendido y la cantidad de dinero entregada por el cliente, calcule e imprima el cambio que se debe entregar al mismo.

Construya un script tal que, dadas la base y la altura de un triángulo, calcule e imprima su superficie. La superficie del triángulo es

$$\frac{\text{base} * \text{altura}}{2}$$

Escriba unscript tal que, dado como datos el nombre de un objeto, su peso y su longitud, expresados estos dos últimos en libras y pies respectivamente; escriba el nombre del objeto, su peso expresado en kilogramos y su longitud expresada en metros.

Construya un script que resuelva el problema que tienen en una gasolinera. Los surtidores de esta registran lo que "surten" en galones, pero el precio de la gasolina está fijado en litros. El algoritmo debe calcular e imprimir lo que hay que cobrarle al cliente.

Construya un algoritmo que calcule e imprima el número de segundos que hay en un determinado número de días.

Dada la matrícula y 5 calificaciones de un alumno obtenidas a lo largo del semestre, construya un algoritmo que imprima la matrícula del alumno y el promedio de sus calificaciones.

